AFRL-RI-RS-TR-2008-247
Final Technical Report
September 2008

AUTOMATED ONTOLOGY ALIGNMENT WITH FUSELETS FOR COMMUNITY OF INTEREST (COI) INTEGRATION

Lockheed Martin Corporation

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2008-247 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/	/s/
JEFFREY W. HUDACK Work Unit Manager	JOSEPH CAMERA,Chief Information & Intelligence Exploitation Division Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
Sep 2008	Final	Jun 07 – Jun 08

4. TITLE AND SUBTITLE
AUTOMATED ONTOLOGY ALIGNMENT WITH FUSELETS FOR COMMUNITY OF INTEREST (COI) INTEGRATION

5a. CONTRACT NUMBER FA8750-07-C-0084
5b. GRANT NUMBER N/A
5c. PROGRAM ELEMENT NUMBER 62702F

6. AUTHOR(S)
James Starz and Joe Roberts

5d. PROJECT NUMBER 558J
5e. TASK NUMBER BA
5f. WORK UNIT NUMBER 03

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Lockheed Martin Corporation
3 Executive Campus
Cherry Hill, NJ 08002-4103

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFRL/RIED
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
N/A

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2008-247

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 88 ABW 08-0078

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Discusses the ontology alignment problem by presenting a tool called Ontrapro—the Ontology Translation Protocol, which allows users to apply a myriad of ontology alignment algorithms in an iterative fashion. This particular work explores the specific cases where a human can augment the capabilities of the machine. The report also discusses situations where the current state of the art in semantic interoperability research can be applied to solve real world problems. Finally we describe operational scenarios that demonstrate the use of Ontrapro/semantic interoperability using new, semi-automatic alignment techniques. These scenarios and lessons learned describe how future work will result in more reliable ontology alignments, further enabling the possibility of semantic interoperability and taking us one step closer towards the original vision of the Semantic Web.

15. SUBJECT TERMS
semantic, interoperability, ontology, alignment, owl, rdf, ontrapro

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Jeffrey W.Hudack
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (*Include area code*)
U	U	U	UU	33	315 330-4348

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Table of Contents

List of Figures

Executive Summary

The future of the Semantic Web envisions an interconnected network of data and systems where software agents can communicate seamlessly to perform complicated tasks with limited human intervention or input. One of the biggest obstacles germane to this vision, however, is the ability of systems to align ontologies correctly to translate and merge disparate but similar domains of knowledge into a single perspective. If ontologies are correctly aligned, the ability to organize and integrate separate data sources enables human or software agents to draw conclusions and gain insight that otherwise would be difficult or impossible. This problem is well recognized by the military and commercial world for having a significant role in today's systems and system of systems. Major software vendors such as BEA and Microsoft offer solutions in this space and many top universities offer approaches to solving the semantic interoperability problem automatically. Unfortunately, both solutions spaces address a small portion of the problem of semantic interoperability.

In this report we discuss the ontology alignment problem by presenting a tool called Ontrapro—the Ontology Translation Protocol, which allows users to apply a myriad of ontology alignment algorithms to the ontology alignment problem in an iterative fashion. This particular work explores the specific cases where a human can augment the capabilities of the machine. Such cases include situations where alignment results are presented for the user to modify and guide the ontology alignment process until an acceptable result set is determined. The report also discusses situations where the current state of the art in semantic interoperability research can be applied to solve real world problems. Finally we describe operational scenarios that demonstrate the use of Ontrapro/semantic interoperability using new, semi-automatic alignment techniques. These scenarios and lessons learned describe how future work will result in more reliable ontology alignments, further enabling the possibility of semantic interoperability and taking us one step closer towards the original vision of the Semantic Web.

1. Introduction

Ontology alignment is a critical aspect of the interoperability between information systems that have varying data semantics. While research in automated semantic alignment has made significant progress in recent years, today's state-of-the-art technology cannot support a solely automated approach to integrate most data systems. Aligning semantics is particularly challenging as it is very dependent on the implicit semantics of the schema, data, and context for integrating the data. Data integration involving multiple ontologies is still a tedious process that must be supported by programmers and database administrators. The time to integrate two complex systems can take years. Additionally, there is little assurance that the new solution will completely leverage the capabilities of the individual systems nor is there a guarantee that the integration will be correct. In fact, it is easy to find anecdotal cases where interoperability led to serious problems, including loss of life, for allied forces.

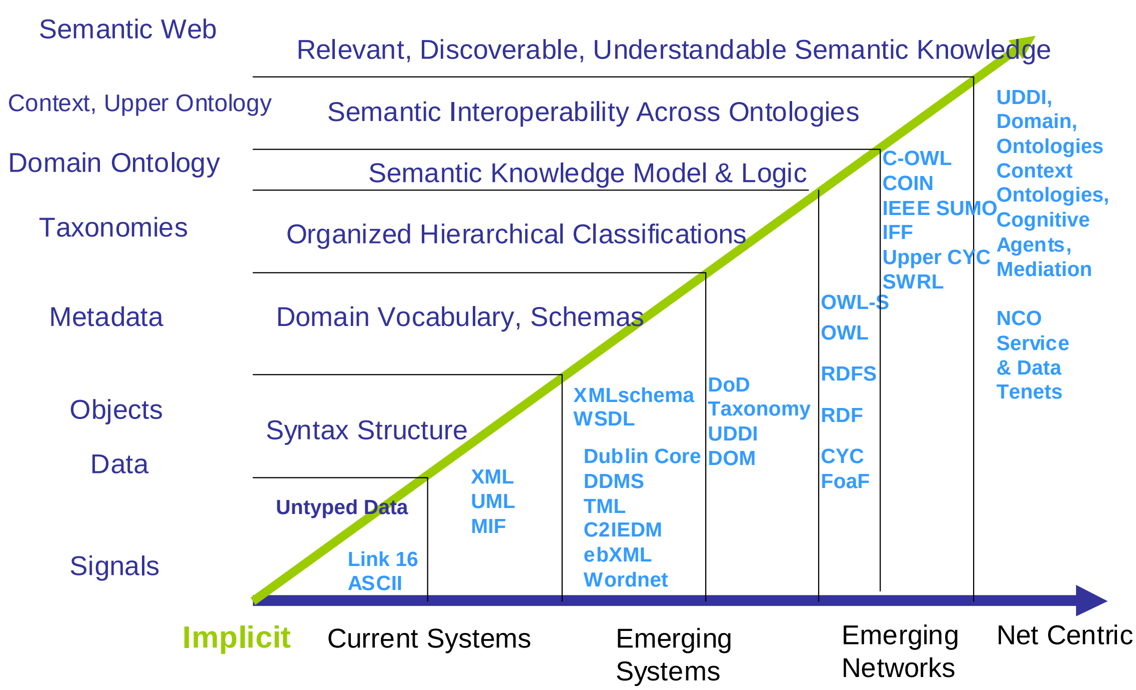

Figure 1 - Semantic Interoperability Perspective [Yanosy]

Ontology alignment involves determining correspondences between similar terms in disparate ontologies or schemas. When systems are integrated, this process is done by a database administrator or a developer. There are commercial tools for aligning schemas, but the task becomes completely daunting as the individual schemas grow. Most research studying this area has focused on automatically aligning ontologies using approaches based on combinations of syntactic similarity, graph similarity, constraint checks, and data analysis.

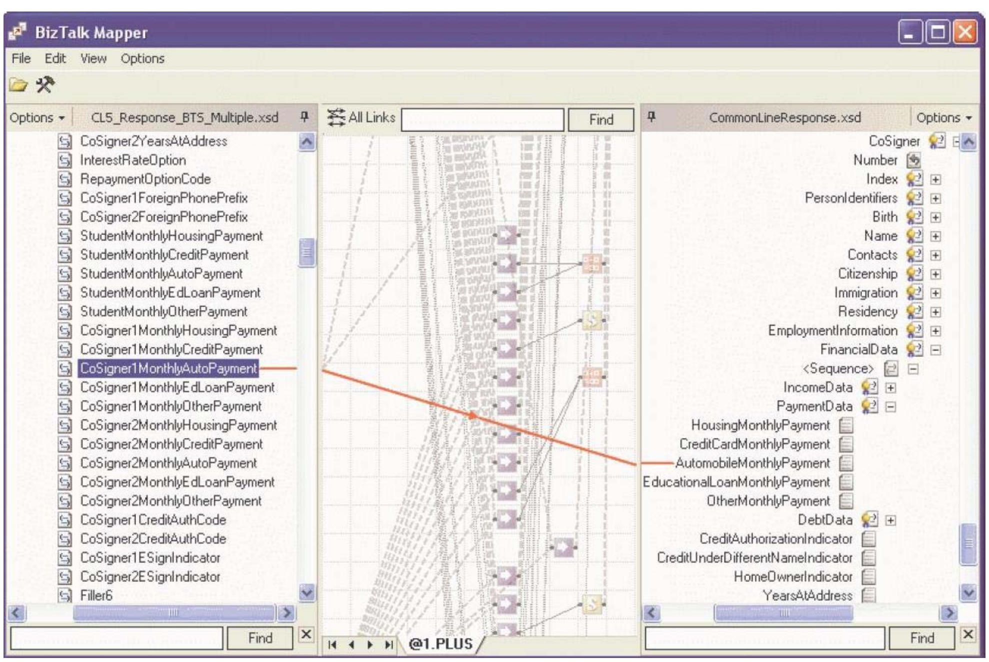

Figure 2 - Complicated Alignment Using MS Biztalk Server

Most ontology alignment algorithms perform some type of linguistic analysis to obtain a preliminary mapping of ontologies. The results from the linguistic analysis phase are often used as a starting point by other analysis methods for further processing. There are many different approaches for linguistic analysis. The simplest method is to calculate a string similarity between the two elements. Strings are assigned an edit value corresponding to the number of operations to transform it from one string into the other. Additionally, lexical analysis can be used to tokenize words which are then compared with similar concept tokens in the other ontology.

Structural matching of elements can be performed based on the similarity of their data structures, context, adjacent elements, and other structural facets. Ontologies are typically modeled using graph data structures during this matching process. Structural analysis assumes that if two elements in different ontological models are found to be similar, the structure of the model can provide insights or hints as to which other elements have a high degree of correlation. In cases where two similar concepts have very little or no string similarity, the analysis of their placement within the structure of the ontology is often the only method to correctly align the two concepts to each other. Analysis methods can vary significantly due to placing more or less emphasis on a variety of structural attributes.

These approaches typically give an incomplete or incorrect set of correspondences between terms. A human must align the remaining terms and check the machine built alignments to truly complete the alignment process. Although fully automated solutions may be infeasible, there are tools and algorithms that, when combined with human assistance, can greatly aid the alignment of large ontologies for which manual alignment is impractical.

These techniques all contain intermediate steps where humans can intervene to manipulate results, parameters, and other data critical to the alignment process. Our work places with an emphasis on exploiting these steps to provide valuable insight to the alignment process and improve accuracy. Meaningful adjustments performed iteratively over the alignment process allow a human user to converge on a significantly more accurate alignment.

Ontology Alignment Overview[1]

Semantic Interoperability refers to the ability of computer systems to exchange information accurately along with the automatic and correct interpretation of the exchanged information by the receiving system. There are a multitude of heterogeneous data sources that exist today, using different ontologies to describe similar domains of knowledge with a high degree of overlap. An ontology can be defined as a formal description of a domain, intended for sharing among different applications, and expressed in a language that can be used for reasoning. The correct alignment of ontologies, therefore, is one of the critical challenges of Semantic Interoperability. For example, an

[1] Additional details can be found in the ATL Ontology Alignment Study Report.

aligner must be able to pair the concepts “car” and “automobile”, as they are semantically similar but syntactically different. This paper will explore the challenges and strategies of ontology alignment, give an introduction to some of the current algorithms in use, introduce Ontrapro as a tool which can be used as a platform to run different algorithms, and touch on other areas of research and interest in the domain of Semantic Interoperability.

The following is a list of issues that ontology alignment faces today along with a short description. Possible approaches to either mitigate or resolve the issue may also be proposed.

This list of current and future challenges was taken from a paper written by Jennifer Sampson from the Norwegian University of Science and Technology [Sampson, 2005].

- Lack of consensus in the literature on terminology - A lack of standardization of concepts and terminology currently exists. Finding similarities between ontologies has been referred to as: ontology mapping, ontology alignment, ontology integration, and ontology merging. Although they are all similar, there are some subtle differences between the terms, which can cause confusion. For example, Ontology integration refers to building a new ontology by reusing existing ontologies and extending and modifying them as seen appropriate. Ontology merging, on the other hand, takes two different ontologies within the same domain and merges them into a single ontology.
- Degree of Automation - Sampson states that the goal of ontology alignment is for automatic alignment of ontologies with no human input or validation, but almost all current techniques require some degree of human input, assessment, and validation. Some debate exists, however, over whether this is a realistic or even desirable goal. With all the subtleties that can exist in the English language, do we really want to automatically align ontologies and not validate the results for possible situations involving life-critical applications where lives can be lost if mistakes are made?
- Challenged in measuring the quality of alignments - No accepted standards for measuring the results of alignments currently exist, and guidelines for evaluating ontology alignment results are needed. Current methods employ the use of human assessment of alignment results to a manually aligned solution which is not realistic because it is prohibitive in terms of required time and effort. Lockheed Martin ATL has proposed an ontology-based approach [Hughes et al] for evaluating alignments in which an alignment confidence rating between 0 and 1 is given for each mapping. Other elements include a field for true and false positives, the number of unaligned elements, and the precision, or proportion of correct alignments found. This proposed standard representational scheme for stating and evaluating alignments in OWL will make it much easier to compare alignment algorithms as well as facilitate greater collaboration among members of the ontology alignment research community.
- Lack of empirical validation using real world ontologies - A scarcity of real world ontologies as well as instances of these ontologies that can be used for empirical

validation of prototype alignment algorithms currently exist. Some potential existing ontologies include the medical ontology Foundational Model of Anatomy (FMA) [Rosse, 2003] as well as the Anatomy Model developed in the OpenGalen Project[2].

- Lack of gold standard ontologies to be used as reference ontologies - Gold standard ontologies as well as alignments between ontologies are needed by researchers to allow the comparison between results of alignment algorithms with the alignments made by human experts. These standard ontologies can also be used as a base ontology which can be extended by other ontologies, naturally resulting in a higher degree of similarity which reduces the complexity of the alignment problem.
- Presentation of alignment results is limited - More research needs to be done as to how to present alignment results effectively in a graphical manner. In the case where large ontologies are aligned, there may be thousands of mappings that need to be displayed effectively without overcrowding the screen real estate. The presentation of results is important because end users often will need to validate or modify automatically generated alignment results.
- Problems with scale and algorithm complexity - Many ontology alignment algorithms experience eroded results when the size or complexity of the involved ontologies increases. The efficiency and performance of these algorithms also suffers. Alignment algorithms must be able to scale efficiently to handle ontologies of all sizes and complexities. Real world ontologies will most likely include thousands of elements containing intricate associations.
- Difficulties in estimating the impact of alignment decisions - Misaligned concepts in banks or medical systems can cause serious errors. When considering the fact that there are no proven automated alignment techniques that can produce results reliable enough for use by critical safety, financial, and medical systems, the risks and rewards must be weighed as to whether or not it is safe to use these techniques.

Alignment Approaches

This section explores some of the general strategies and approaches that ontology alignment algorithms use towards the goal of semantic interoperability. The concepts presented in this section are intended to provide a very high level introduction to different types of approaches taken, as there naturally exist many types of variations to these concepts due to the diversity of algorithms available.

Wordnet

Wordnet is a lexical database of the English Language where English words are grouped into sets of synonyms called synsets. The database can then be used to support automated text analysis and natural language processing. More specifically, alignment algorithms can be used to look up synonyms for similarity calculations for semantic and lexical analysis purposes. Debate exists about the usefulness and efficacy in using Wordnet or a thesaurus in ontology alignment. Accessing the Wordnet database in search

[2] http://www.opengalen.org/

of synonyms is time-consuming and has not been empirically shown to produce better results in alignment results. Problems with the efficiency of algorithms employing the use of Wordnet have also been raised. Further investigation and experimentation should be done to ascertain the efficacy of using Wordnet in ontology alignment.

SENSUS is a 70,000 node terminology taxonomy which is an extension and reorganization of Wordnet. It serves as a framework into which additional knowledge can be placed. Its stated goal is to provide a wide-ranging semantic thesaurus that is built incrementally which can be used by reasoning/inference engines for a deeper semantic understanding of texts.

A possible area of research would then involve the construction of a military version of Wordnet where a lexical database of military terms could be categorized and grouped to aid in ontology alignment of military systems. This idea would probably only be explored further, however, if Wordnet/SENSUS is shown to have a significant impact on the efficacy of ontology alignment results in the algorithms presented in this paper.

Linguistic Analysis

Almost all ontology alignment algorithms perform some type of linguistic analysis to obtain at least a preliminary mapping of ontologies. The results from the linguistic analysis phase are often then used as an initial mapping by other analysis methods for further processing.

There are many different approaches for linguistic analysis. The simplest method is to calculate a string similarity between the two elements. For example, the elements ‘Dept’ and ‘DeptNo’ would have a similarity value of 0.66. String similarity can also be thought of in terms of edit distance. The edit distance is the number of operations required to transform one string into another. There are different methods to calculating the edit distance, such as the Levenshtein method or the Jaro-Winkler method. Using the Levenshtein distance method, the distance between “kitten” and “sitting” would be 3.

1. kitten → sitten (substitution of 's' for 'k')
2. sitten → sittin (substitution of 'i' for 'e')
3. sittin → sitting (insert 'g' at the end)

Lexical analysis, or the breaking up of the element into tokens is also another approach often used during the linguistic analysis phase. The tokens can then be used individually to help in the matching process, often by finding concept tokens in the other ontology that are similar.

Obviously the results of simple string comparisons are very rough and preliminary and can often give misleading results, but again its results are useful to use as an initial mapping or starting point for more complicated analytical methods.

Structural Analysis

Many alignment algorithms also perform structural matching of elements based on the similarity of their data structures, context, adjacent elements, etc. Structural analysis can be performed by representing the ontology as a model, directed label graph, tree, any other data structure. The motivation behind structural analysis is the assumption that if two elements in different ontological models are found to be similar, the similarity of their neighboring elements also increases. In cases where two similar concepts have very little or no string similarity, the analysis of their placement within the structure of the ontology is often the only method to correctly align the two concepts to each other. Structural analysis, therefore, is often the critical aspect in an alignment algorithm in determining its efficacy since linguistic analysis alone is generally insufficient.

There are many variations on the type of structural analysis performed. For example, some algorithms like Cupid put more emphasis on atomic elements or leaves in a tree. Similarity Flooding [Melnik et al, 2002] runs its structural analysis algorithm over many iterations on its model graph, and assumes the initial similarity of two nodes will propagate through the graph until a fixpoint is reached. Some algorithms, such as ASCO and OLA do not utilize neighboring information at all. Other algorithms take advantage of the structured organization of RDF and OWL in performing their similarity analysis. More details will be given in subsequent sections in which the individual algorithms will be examined.

Human Analysis

The majority of ontology alignment algorithms proposed thus far are designed to be semi-automatic. That is, intermediate steps exist where humans can tweak the current results to their liking or set other parameters or heuristics as seen fit. Many approaches aim to present humans with only a “best-guess” solution of the alignment, and require the human to parse the results and manually make modifications before accepting the final alignment. In these cases, the semi-automatic algorithms are only seen as an aid to simplify the alignment problem, since the original ontologies are so large that it precludes the possibility of manual alignment. As stated in the previous section, the ultimate goal is a fully automated alignment process where human intervention and analysis is non-existent. Debate exists, however, over whether this is a realistic or even a desirable achievement, especially in the previous stated case where life-critical operations are dependent on the results.

Alignment Algorithms

This section will provide a quick introduction to the variety of alignment algorithms that exist in the research domain today. This section should illustrate to the reader the wide variety of approaches taken towards ontology alignment. A complete list of alignment algorithms can be found in the ATL Ontology Alignment Study.

Anchor-PROMPT

Anchor-PROMPT [Noy et al, 2001] take as its input a set of related pairs called anchors from the source ontologies. These anchors can either be identified by the user manually or the system can identify them through lexical analysis. By using the set of anchors,

Anchor-PROMPT can identify new pairs of semantically close terms. This is accomplished by traversing the paths between the anchors and incrementing the similarity score between the elements that are reached in the same step. This process is repeated for all possible paths that can originate and terminate at the anchor points. The reasoning behind this strategy is that if there are two pairs of terms that are known to be similar, then the paths that connect the terms contain elements that are also similar. A small set of identical terms, therefore, can result in a large number of terms that are also semantically similar.

Cupid

Cupid [Madhavan et al, 2001] is an algorithm that uses both linguistic and structural matching techniques, taking a weighted average for the resulting final similarity value.

During the linguistic matching phase, a normalization step uses tokenization, expansion (identifying abbreviations/acronyms), and elimination (discarding prepositions, articles, etc) to process the data. Elements are then separately clustered into categories. Linguistic similarities are then computed between elements by comparing the normalized tokens, using substring matching along with the help of a thesaurus to determine synonymy and hyponymy relationships. The resulting similarity is called the linguistic similarity coefficient.

The structural matching phase is based on the similarity of the element's contexts or vicinities. A tree data-structure is used, and the basic premise is that atomic elements, or leaves, in two trees are similar if they are linguistically similar or similar in data-type, AND elements in their vicinities (ancestors and siblings) are also similar. Non-leaf elements are also considered similar if their subtrees are similar.

The resulting similarity is called the structural similarity coefficient. After these two phases are completed, both the linguistic similarity and structural similarity coefficients are averaged together to produce the final similarity coefficient.

OWL-Lite Alignment (OLA)

OLA [Euzenat et al, 2005] is an algorithm in which both string distance and lexical distances are computed for the comparison between Universal Resource Identifier References (URIrefs). The algorithm is designed for alignment of ontologies expressed in OWL. The lexical distance computation relies on WordNet for a quantitative assessment of the similarity between the two terms. OLA currently does not consider inheritance in its alignment processing out of efficiency considerations. OLA constructs an OL-Graph, which is a labeled graph where vertices correspond to OWL entities and edges to inter-entity relationships. The similarity value of two nodes then depends on the similarities of the terms used to designate them, the similarity of the pairs of their neighbor nodes linked by edges expressing the same relationships, and the similarity of other features such as cardinality and property types.

Google Distance

This algorithm introduces the new concept of using a Google-based similarity measure as a heuristic to minimize the "sloppiness" required for desirable matches, while maximizing the "sloppiness" required for undesirable matches [Gligorov et al, 2007]. Sloppiness is the concept that a fraction of the submappings in a mapping can be ignored. A high sloppiness value will in turn allow mappings between any two arbitrary concepts, even when there is no real degree of correspondence. A potentially significant amount of incorrect mappings, therefore, would exist using high sloppiness values. Using the Google heuristic weighting function would help ensure that when the allowed sloppiness level is slowly increased, desirable matches are quickly found at low sloppiness values, while undesirable matches are only discovered late in the process when the sloppiness value is very high. The gradual increase of the sloppiness value results in an early increase of recall, but a late decrease of precision. A dissimilarity measure called the Normalized Google Distance (NGD) is used. NGD uses the number of hits returned by Google to calculate a semantic distance between concepts. By using this measure, it provides a measure of the probability of the co-occurrence of term y within the same web page that includes a term x. The probabilities, or weights, are then used in calculations of the sloppiness value to determine whether or not the match is desirable.

GLUE

GLUE [Doan et al, 2002] is an algorithm which matches taxonomies using machine learning techniques to find mappings. GLUE is unique in that it is flexible and scalable to support the use of multiple learning strategies. This is of particular interest because the algorithm can contract or expand based on a combination of differing learning strategies that are deployed, which may create a whole new field of possible research as to which combination of strategies are most effective. Each of these strategies would take a different approach on how to process the data or the taxonomic structure of the ontologies. The predictions from the set of learners are combined by a meta-learner for a unified solution. GLUE's approach to measuring similarity is unique because it is based on the joint probability distribution of the concepts involved. This joint distribution is used by the learners to compute its suitable similarity measure. For two concepts A and B, the joint distribution consists of 4 values:

1. Probability that an instance in the domain belongs to both A and B
2. Probability that an instance in the domain belongs to A but not to B
3. Probability that an instance in the domain belongs to B but not to A
4. Probability that an instance in the domain belongs to neither A or B

Based on the joint probability distribution, the Jaccard Coefficient[3] is derived. The Jaccard Coefficient is a measure of similarity between the two sample sets.

GLUE also purports to incorporate common sense knowledge and domain constraints into the matching process. This is done by using general heuristics to improve mapping accuracy. For example, one heuristic is that two nodes are likely to match if nodes in

[3] http://en.wikipedia.org/wiki/Jaccard_index

their neighborhood match. Relaxation labeling is a powerful technique that is used to effectively incorporate and handle all the heuristics and domain constraints used.

GLUE works by taking in two ontologies along with their data instances, and computes the joint probability distributions using machine learning techniques. The results are fed into a similarity estimator which applies a user-supplied similarity function to compute a similarity value for each pair of concepts. The output is then a similarity matrix, which is used by a relaxation labeler to apply domain-specific constraints and heuristics to find the best mapping configuration which best satisfies the constraints. This mapping configuration is then the final output of GLUE.

Virtual Documents

Virtual documents [Qu et al, 2006] are documents for which no persistent state exists and for which some or all instances are generated at run time. In terms of ontology alignment, a virtual document is a collection of weighted words. A virtual document is generated for each URIref declared in an OWL/RDF ontology. The unique quality of virtual documents is that a virtual document of an URIref contains not only the local descriptions but also the neighboring information that affects the meaning of the URIref. A weighting schema is also used to reflect the importance of the information. Experiments have shown that combining virtual documents with the TF/IDF technique described earlier in the ASCO algorithm resulted in effective linguistic matching for ontologies.

Virtual documents are represented by a collection of weighted tokens (or words), where the weights are rational numbers. These tokens are generated through a pre-processing of the ontology where the document is broken up into words weighted to indicate their importance within the document. Non-content bearing words are eliminated during this process.

For each URIref, iteration equations are applied until a convergence solution is reached. Usually 5 iterations are computed before convergence occurs. Descriptions of neighbors are included in virtual documents by using neighboring operations to describe different types of neighbors. Note that RDF triples or written in the order (subject, predicate, object). Therefore, the neighbor types are the all nodes SN(e) that occur in triples with an URIref denoted by e as the subject, all nodes PN(e) where e is the predicate, and all nodes ON(e) where e is the object. A collective function is then computed for each URIref virtual document, using all the neighbor types as well as the collection of works/tokens in each URIref to calculate the final weight.

The similarity weight calculated for the virtual documents is then combined with the TF/IDF technique to form a final similarity score between 0.0 and 1.0.

Distributed Description Logic (DDL)

DDL [Meilicke et al] differs from all the other algorithms in this section because it is actually not a matching algorithm, but a tool that can be used to improve mappings using logical reasoning. Therefore, it is actually orthogonal to any matching algorithm and can

be used in combination with any matching algorithm to weed out and incorrect mapping results. By using a set of rules, DDL can determine confidence values for each mapping, analyzing the impact of the created mappings on the ontologies, and eliminate mappings that have a malicious influence.

The rules will help create irreducible conflict sets in the system. An irreducible conflict set is a set of mappings that make the concept unsatisfiable. The removal of a mapping, however, will make the concept satisfiable again, therefore indicating that the mapping should also then be removed from the final result set.

Research into the efficacy of DDL can be performed by applying this tool to results of mapping results from any of the alignment algorithms described in this document and displaying the results in the proposed OWL alignment result format proposed by Lockheed Martin ATL. This would result in a quick and efficient assessment of the efficacy of DDL because there will be fields for statistical measures such as false negatives, correct matches, etc. that provide an unbiased analysis of whether or not beneficial changes were made in the mapping results.

Structure-based filtering

The structure-based filtering approach [Chen et al, 2006] contrasts from other approaches in that structural information is used only as a filtering method to remove wrong results, but not for the computation of the similarity values between terms. This philosophy is based on the assertion that using information about the structure if ontologies has not produced good results for alignments, but could be helpful in filtering out wrong results. This approach is also unique in that two similarity thresholds are used. The lower and upper thresholds separate all matchings into 3 categories: Pairs above the higher threshold; Pairs between the higher and lower thresholds; Pairs below the lower threshold. In the structure-based filtering algorithm, all pairs above the higher threshold are assumed to be a valid match. Pairs below the lower threshold are automatically discarded. Finally, pairs between the 2 thresholds are analyzed using structural filtering to ascertain if they will be retained. This approach is also flexible because it is not married to a single matching approach and would allow the use of different matchers to calculate the similarity scores.

Once a matcher has been used to calculate similarity scores and the partitioned groups have been set, a consistent suggestion group is calculated. Consistent suggestion groups are matches that are consistent to each other with respect to the structure of the ontologies. These matches are derived only from the partition of matches that have a higher similarity score than the higher similarity threshold used to create the partitions. A match is part of a consistent suggestion group if each match occurs at most once in a first argument in a pair, at most once as a second argument in a pair.

The consistent suggestion groups are then used to partition the original ontologies into 3 separate parts. For an element A, the groups are divided into the descendants of A, the element A itself, and all others. This partitioning is done for all members in the

consistent suggestion group. The following figure shows the partitioned ontologies for the consistent suggestion group {(2,B), (3,F), (6,D)}.

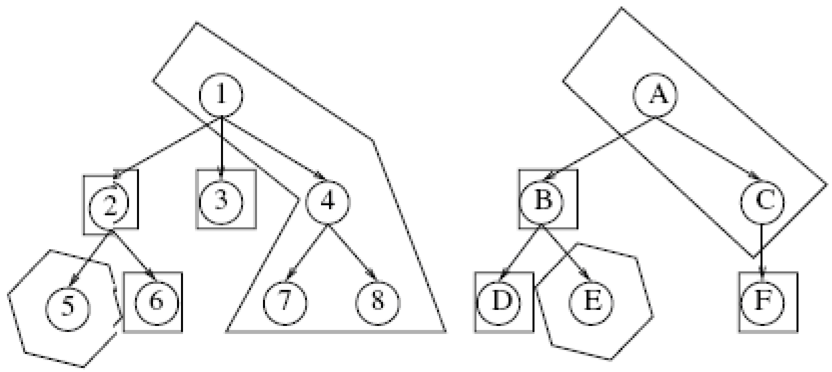

Figure 3 - Partitioned Ontologies

Finally, all pairs with similarity values in between the higher and lower thresholds are evaluated and filtered using the partitioned ontologies. Pairs in which both elements belong to the same partitioned group in the ontologies are considered viable matches, and all others are discarded. For example, (5,E) would be a valid match, while (5,C) would be discarded. The final ontology alignment will include all matches with a similarity value greater than the upper threshold as well as all filtered matches with similarity scores between the higher and lower thresholds.

Fragment Oriented Matching

Fragment oriented matching [Rahm et al, 2004] is an approach to ontology alignment where a large match problem is broken up into several small ones, and reusing previous match results to help in matching new fragments. The reasoning behind the strategy for this divide-and-conquer approach is that the effectiveness of many automatic matching techniques experiences a significant decrease in performance when the input ontologies or schemas are large because of the greater possibilities of false matches. By breaking up the matching problem into fragments, this approach is extremely scalable and therefore capable of handling alignment problems of all sizes.

The fragment-based match strategy is composed of 4 steps:
1) A decomposition step to determine suitable fragments
2) Identification of the most similar fragments between schemas to match
3) Matching similar fragments
4) Combining the fragment match results

A fragment is defined in this context as a rooted sub-graph in the schema graph. In the paper, XSD schemas are used as the primary example. Therefore, sub-schemas, which can be separately instantiated, schema nodes, and entire schemas themselves, can all be considered a fragment. The goal is to have as little overlap as possible between fragments to try to avoid un-necessary repeated computations as well as overlapping results. Fragments are then paired together by examining their metadata, contexts, names, etc to try to associate fragments from two different schemas that have some

degree of similarity. Finally, the paired fragments are matched using different selected techniques such as name or structural matching. During this final matching phase, previous match results may possibly be reused, due to the assertion that the reuse of match results are more applicable at a fragment-level as compared to entire schemas.

General Observations

The list of algorithms that have been introduced in this section should illustrate the wide variety of approaches and strategies taken towards the goal of successful ontology alignment. From simple string comparisons to complex mathematical computations, many methods exist to calculate linguistic similarities, and the methods for ascertaining structural similarity of concepts in ontologies are just as diverse.

In the next section, we introduce a tool developed by Lockheed Martin Advanced Technology Laboratories which can be used to run and test a majority of the algorithms introduced in this paper. Using an intuitive user interface, users can quickly compare and test a wide range of proposed solutions to the ontology alignment program on standard sets of data.

ONTRAPRO

Ontrapro is a tool developed by Lockheed Martin Advanced Technology Laboratories to automatically discover semantic correspondences between heterogeneous data models with no set explicit mappings. The extensible software architecture of Ontrapro allows for the integration of a variety of ontology alignment algorithms and approaches. Ontrapro is capable of comparing syntactical, lexical, and structural components between data models to identify the widest range of semantic similarities. Ontrapro currently implements the capability to apply the Similarity Flooding [Melnik] and Anchor-PROMPT [Noy] alignment algorithms to disparate sets of ontologies. A Graphical User Interface (GUI) was built to simplify the alignment process, allowing the user to select which algorithm to use and what ontologies to align. A result pane displays the initial results in a Notation3 format, which is a shorthand non-XML serialization of a Resource Description Framework (RDF) or Web Ontology (OWL) model in a more human-readable format.

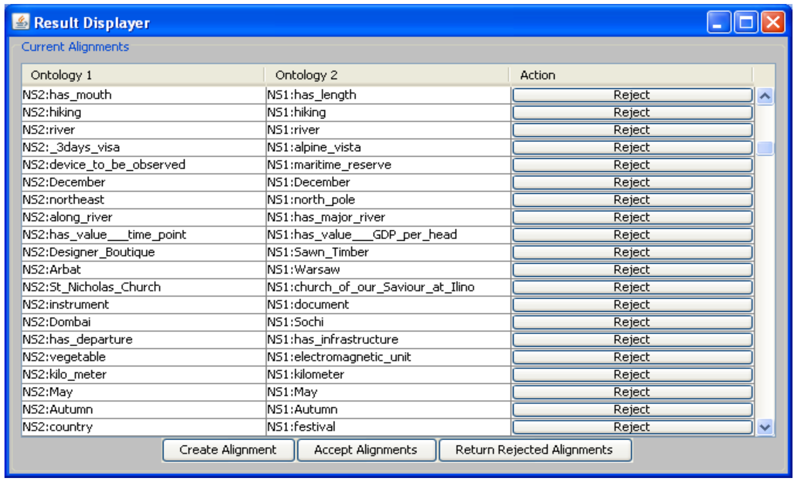

Figure 4 - Ontrapro Alignment Results Display

Fuselet Technology

Today, information transformation is an activity performed in a disjointed, ad-hoc manner. To the extent that these efforts succeed in providing useful information to some consumer, this is not necessarily a problem. But there is a substantial, untapped potential in today's information management environments to apply shared information transformation components in a managed infrastructure in order to provide a transformation capability that is more reliable, repeatable, scalable, measurable and manageable.

Fuselet technology provides these benefits by offering distributed containers capable of executing and controlling transformation components built from reusable, parameterizable software components. Implementing transformations with Fuselets is:

- **Reliable**. By creating transformations from reusable, parameterizable components rather than ad-hoc scripts, transformation logic is much less likely to contain errors. By running transformations in a managed container, problems with ongoing transformations are much more likely to be detected via logging and alerting features and therefore to be corrected in a timely fashion.
- **Repeatable**. Not only can fuselets be created from reusable components, but fuselets themselves provide "reusable" information insofar as their outputs are delivered via publication, allowing many information consumers, including other fuselets, to concurrently utilize the results of a fuselet transformation. This reuse of logic and results makes for much more repeatable information production than that of many clients each creating their own custom, one-time transformations.
- **Scalable**. Many transformations will be useful to many information consumers. By running shared transformation components, significant savings in both computational and communications resources are possible, allowing both lower utilization and higher numbers of transformations.

- **Measurable**. Running transformation components in a container allows us to measure the runtime performance characteristics of fuselets and populations of fuselets and also to measure and log aspects of the results of their transformations, for further analysis and refinement of the transformation logic.
- **Manageable**. A managed container allows us to control the operation of transformations both in aggregate and in a fine-grained manner. Malicious or malfunctioning fuselets can be limited or shutdown with both automated and manual mechanisms. Furthermore, but running fuselets within an overall information management environment, organizational and system policies can be applied to fuselets, including security, configuration, and prioritization policies.

SI Research

Overview

The initial design of the system was based on having an information management staff member use Ontrapro to help build fuselets capable of translating messages between other messages. Much of this infrastructure was used on all three prototypes, with the relaxation of the staff involvement being the major change.

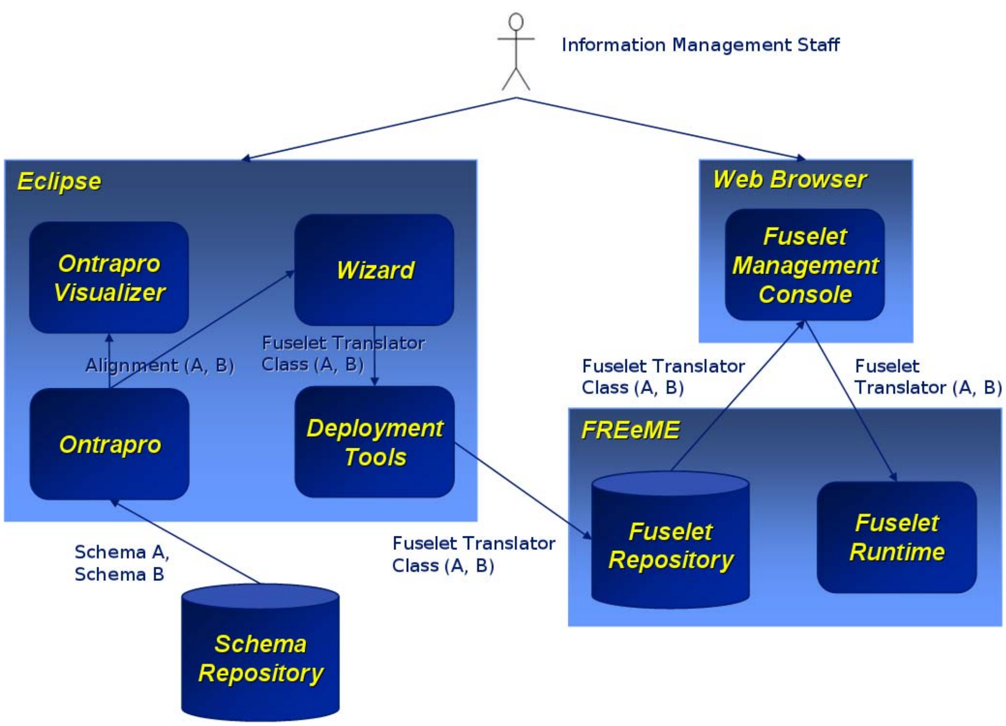

Figure 5 - Initial Component Flow

Our technical approach towards the ontology alignment problem centers on augmenting the original Ontrapro code to allow user-guided input and provide the capability to run multiple iterations of the alignment algorithms as needed until an acceptable result set is

determined. In the majority of ontology alignment scenarios, the user will have little knowledge of the contents of at least one of the ontologies they desire to align. Therefore, instead of querying the user to identify alignments that they know beforehand to be correct, we instead provide an initial result set for the user to analyze. The user can then reject alignments they determine to be incorrect. All rejected alignments are submitted back into Ontrapro to help guide the alignment process and mold the subsequent result set. This is accomplished by setting the alignment rating for each rejected pair in the sparse matrix data structure used to represent alignments to zero. Each completed iteration will present an original result set to the user. In each set, original alignments are suggested for elements that had been previously misaligned. All non-rejected alignments are implicitly assumed to be correct and remain the same in the new alignment set. A database of rejected alignments is also stored to ensure that previously identified misaligned elements are not presented again in future iterations to the user. Although this approach requires a moderate user degree of involvement to confirm or reject each alignment suggestion, the matching effort to align the elements is still automated and relieves the user of the task of determining correct alignments manually.

Many ontology algorithms, however, do not need the user to manually determine correct alignments as a necessary prerequisite for their algorithm to execute. On the other hand, a sizable percentage of ontology alignment algorithms and tools present a single result set to the user with very little or no user input. Given the highly subjective nature of ontology alignment and the strong probability or even near certainty that incorrect or sub-optimal alignments exist, some mechanism of obtaining user feedback should be available. Ontrapro's user feedback system is pertinent to the average user because it is generally within their realm of expertise to be able to identify at the minimum grossly misaligned elements. Alignments that are not rejected are assumed to be correct but can always later be rejected in a future iteration. Ontrapro's user feedback system also benefits users with expertise in the ontologies they align because it allows the user to manually specify an alignment that they know to be correct. The Result Displayer takes the original output of Ontrapro in Notation3 format and displays the data in a user-friendly table. Users have the capability to manually modify the text fields containing the alignments. Another newly engineered capability of Ontrapro is the ability to display the unaligned elements for each ontology after every iteration and is illustrated in the figure below. This optional capability allows users to view which classes in the ontology were not aligned and can be useful in scenarios where ontologies need to be merged and unique classes from each ontology may have to be included in the final merged ontology to further add semantic value. A final and critical advantage to our technical approach is that as long as users do not accidentally reject any correctly aligned elements, each iteration in nearly all cases will at worst produce an equally precise result set of alignments. Over the course of the alignment process, result sets will produce more precise alignments and fewer false positives after the completion of each iteration. A precise alignment set is critical towards fulfilling the vision of the Semantic Web, where data can be integrated and used across various applications.

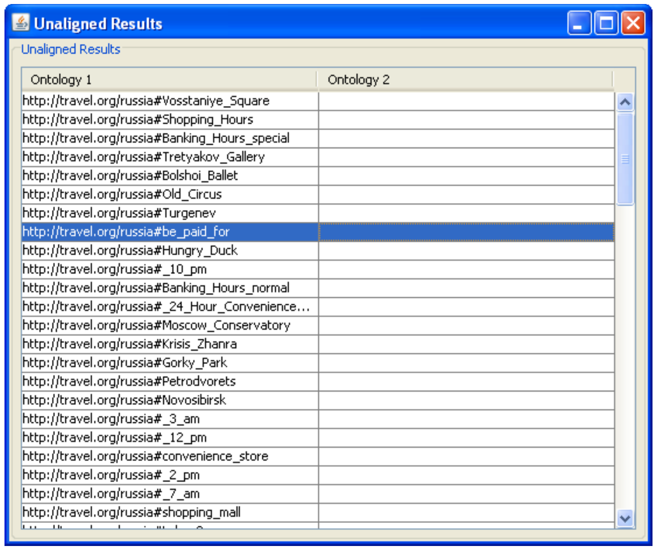

Figure 6 - Unaligned Results

Experimental Results[4]

For the purposes of our experiments, the choice was made to only apply the Similarity Flooding algorithm towards our sets of ontology data, although Ontrapro is also capable of executing the Anchor-PROMPT algorithm. This is because Anchor-PROMPT suggests new alignments based on a provided list of correct alignments, or anchors. No correct alignments are known in the beginning of our experiments. Also, the current implementation of Anchor-PROMPT in Ontrapro only suggests new alignment results and do not contain previously implicitly assumed correct alignments which are necessary for our iterative approach. Ontrapro allows the user, however, to input the results from Similarity Flooding into Anchor-PROMPT as de facto anchors to generate even more original alignment suggestions.

To test the efficacy and validity of our technical approach, we used two sets of fully developed ontologies from the Ontology Alignment Evaluation Initiative (OAEI)[5], an organization which organizes campaigns and contests aimed at evaluating ontology matching technologies. These ontologies were used in previous contests as standard sets to evaluate the correctness of alignment results of a variety of ontology alignment approaches. The specific ontologies that were chosen were the russia1.owl, russia2.owl, sportEvent.owl, and sportSoccer.owl ontologies. These ontologies were chosen because full and correct alignment results exist between the ontologies enabling us to correctly calculate our alignment precision. The ontologies were also extremely large, precluding any reasonable efforts of manual alignment in a real-world setting.

For our experiment, we decided to run through five alignment iterations for each set of ontologies. We believe that this was the minimum number of iterations that should at

---

[4] Full experimental results can be found in Danny Chen, John Lastusky, Jim Starz, and Steve Hookway. User Guided Iterative Alignment Approach for Ontology Mapping., SWWS 2008.

[5] http://www.ontologymatching.org

least demonstrate some minute level of improvement in our alignment accuracy. After each iteration, we rejected each alignment that was incorrect, based on the correct matching results that were provided by the OAEI.

After five iterations, 14.16% and 18.39% additional correct alignments were found in the Russian and sport ontologies, respectively, with an average of 17.39% additional correct alignments when combining the results. For the Russian ontologies, the first iteration (before any user-guided input) found 70.19% of the total correct alignments. After five iterations, the number had increased to 80.12%. For the sports ontologies, the first iteration found 47.33% of all correct alignments, and after five iterations 58% of all correct alignments had been found. In conclusion, the data that has been provided supports the claim that our iterative approach towards ontology alignment results in alignment sets of increasing accuracy.

An interesting parallel can also be drawn with our assertions and findings with the incremental schema matching approach proposed by Microsoft Research Labs [Bernstein et al, 2006]. Like our iterative alignment approach, Microsoft's incremental schema matching proposes a method to negate false positives and avoid many of the frustrations of ontology alignment, including the inability to see second and third choices. They also reject the idea of a single shot approach towards alignment of data models and demonstrate a tool that integrates human intelligence with machine reasoning to produce a final schema mapping. We believe that our research has supported the findings of the work originally performed by Microsoft Research Labs by objectively demonstrating the effectiveness of an iterative approach that allows a user to reject any false alignments, align elements to originally sub-optimal matches which actually are correct, and play a greater role in the determination of the final matching set.

A few subtle distinctions, however, exist between the strategies behind and the presentation of our similar approaches towards the ontology alignment problem. In this paper, statistical and objective data is presented to support our assertion that an iterative alignment approach can produce better alignment results when compared to some single shot techniques. These approaches can mitigate some of the inconveniences inherent in single shot alignment techniques mentioned earlier in this section. This highlights the potential of and the need for deeper and more substantial research into incremental and iterative approaches in the field of ontology alignment. Another distinction is that our approach is more heuristics-based in nature when compared to Microsoft's approach, which is more involved because the user must highlight each individual element and press a hotkey to display suggested alignments. There are pros and cons to both methods, depending on the user and their preferences. If the user must generate an alignment quickly to come up with a best guess solution and tailor the results from that point on, Ontrapro would be able to fulfill those requirements. If the user requires a very finely-tuned alignment and needs to take advantage of their expertise in the domains of interest represented by the ontologies, Microsoft's incremental schema matching approach may be better suited for that purpose. Finally, if the ontologies are very large in scale, such as the examples used in our experimental scenarios consisting of hundreds of elements, it may not be realistic to use their incremental schema matching approach because of the

time and effort costs involved. In this case, using a heuristic approach makes more sense. Both methods, nonetheless, present value-added contributions in the pursuit of stronger solutions for the ontology and schema matching problem domains.

Our results show that the Ontrapro tool is an important tool in the continual pursuit of stronger and more robust ontology alignment solutions. Ontrapro's main contributions are its ability to iteratively apply the Similarity Flooding algorithm towards a set of data allowing the user to mold the final alignment set to maximize the accuracy of the final alignment, its ability to execute different algorithms to a standard set of data, and the architectural framework that it provides to easily integrate cutting edge alignment algorithms conceived by the research community. While fully accurate and automated alignment solutions are beyond the reach of current technologies, it is possible to provide "good enough" alignment results with minimal human interaction. More importantly, it is possible to generate useful results without intimate knowledge of the merging ontologies.

Although we believe our user-guided iterative approach towards ontology alignment is an exciting development with high potential, there are some limitations and risks inherent in our approach. Ontrapro currently is an application that is still in the prototype stage; the development process of Ontrapro is still ongoing, but the features and capabilities that it provides demonstrate its potential as the system evolves into a production-grade application. Research to explore alternate and more user-friendly methods is progressing, allowing the user to enter input resulting in an alignment set with maximum precision. For example, some of the work currently being performed allows the user to view and select one of the next three best scored alignments to reduce the total number of iterations required for a satisfactory result set. We are also adding the capability to color code alignment results based upon their confidence ratings.

Our experimental approach also relied on the possession of exact matching results so that the correct alignments can be selected for rejection to maximize the accuracy of the result set. In a real operational scenario, exact matching results will not exist and it will be difficult to ascertain the stage of maturity of the alignment set. Performing a fixed number of iterations on a set of ontologies does not guarantee any level of precision, although in almost all cases the user can be reasonably confident that the current result set will be more precise than in the past.

The ripple effect is also highlighted to demonstrate some potential limitations of our iterative approach. A positive or negative mapping will have a ripple effect on the other existing mappings. For example, if "nickname" is incorrectly mapped to "last_name" and is never rejected by the user, no number of iterations performed will produce the correct alignment of "last_name" to "family_name". If two elements are correctly aligned, this will also have a ripple effect on the resulting mappings since these elements will be removed from the pool of consideration for alignments. A smaller domain of potential alignments can possibly result in fewer iterations for a higher level of accuracy in the alignment mappings. In conclusion, the ripple effect can have a subtle yet potentially dramatic impact on the final result set.

Finally, the task of comparing and rejecting alignments is menial and error-prone, especially for large ontologies and schemas. The current implementation of Ontrapro precludes the possibility to undo rejected alignments. Work is currently ongoing to highlight previously identified correct alignments from prior iterations so the user's attention is focused on the newly suggested alignments to hasten their evaluation.

Research Conclusions

With the explosion of data on the web, the challenges and need for ontology alignment is apparent. Our results demonstrate that while ontology alignment is a difficult problem for humans, the process can be automated enough to provide meaningful information in a decision making process with minimal human interaction. Although a human user is required to finish the alignment process, there are techniques, while still experimental, that can effectively reduce the amount of arduous work a user must perform. Specifically, we have shown methods to complement the human activities with machine capabilities to get value from each of their unique qualities. These advancements give us hope that the future of a functional Semantic Web may be within our grasp.

Demonstration/Vignettes

To demonstrate the capabilities of semantic interoperability we focused on three different demonstration vignettes. The three were meant to show various features of semantic interoperability problems and were all byproducts of the natural flow of the effort.

Improvisational Integration

The first demonstration thread developed focused on integrating information from disparate data source into a centralized consumer system. For the sake of convenience, we choose to leverage INTERACT, a collaboration environment built by Lockheed Martin, as a centerpiece of the integration. The hypothesis was that non-programmers could be aided by semantic interoperability technologies to add a new data source into their exploitation or visualization system.

The challenge is very evident in today's system of system environment used widely throughout the military. There are myriads of complicated command and control systems along with many specialized applications and data feeds. To integrate said data sources can require a formal process that can easily take months to complete. A useful capability of semantic interoperability technologies would be to support the integration of new/pop-up information sources on demand. This would provide our military with significant advantages in terms of speed and information superiority. Given this capability does not exist today, we needed to pinpoint the areas where semantic interoperability could help address problems.

To investigate this problem, we choose to leverage Lockheed Martin's INTERACT collaboration software. This software leverages various data feeds that can easily be displayed on a map and shared information spaces. This tool has a lot of similarities to Command Post of the Future or FalconView that both contain a significant number of data feeds that make the utility of the system possible. In INTERACT, there has been

significant activity involved with adding data sources to act as information feeds that can be displayed. This activity was always done by a developer.

At the most basic level, the ability to add information to INTERACT is fairly straightforward. There are web-friendly APIs and intuitive use of geospatial objects. However, there was no obvious way to bridge the gap from a developer to a more common end user. The decision was made to build a mash-up like capability that was akin to what is available on the web. It provided the capability to add simple information objects while constraining the end-user from doing much more.

In the actual demonstration scenario, there is already existing information and ontologies in the INTERACT system. There is a new information source related to a disaster that has just occurred. In this problem, the need to quickly integrate information is paramount. The application demonstrated allows users to map a few of the key fields from the native data source into the INTERACT data model. For display purposes, only three fields are required, latitude, longitude, and name. Not only are there few fields, but it is also likely that these three fields could be mapped automatically. The low barrier to add information makes it possible for the end-user to perform this process very quickly. It also allows them to easily add additional information to the alignment. Though much of the integration capabilities were geared for INTERACT, a looser coupling to the end-application may be possible. Additionally, it is possible that such ties to a particular end application can be done without significant programming.

The takeaway lesson from the Improvisational Integration thread is that there is likely a space of tools between the mash-up toolkit found on the web and the commercial schema matching tools available by commercial vendors. In the INTERACT case, we found that the more assumptions we could make while building the application, the simpler the wrapping process would be. That said it isn't a far leap to imagine a situation where end-users could take these tools and integrate information on the fly.

Multiple Source Query

The second demonstration thread developed focused on integrating information from various sources through a single query. This is the traditional federated search problem, where the sources don't necessarily adhere to a common schema. This is a very legitimate situation. Data is often stored in redundant, similar, or a conflicting manner. The hypothesis is that you can perform multiple source querying using automated alignment by relaxing some constraints and leveraging the expertise of the user.

The challenge is pervasive in today's information space. There are nearly always heterogeneous sources of overlapping information. There will neither be a unifying schema that will apply to all the sources, nor sufficient time to manually build up alignments at query time. The belief is that you could leverage information about the user's information needs and about the query to guide the actual integration effort.

The following example shows a user making a simple request from three data sources for information concerning automobiles. While the information requested in this case is

contrived, this simple scenario demonstrates a fraction of the challenges of Semantic Interoperability and how our system will attempt to address them.

Consider a user's need to find all cars and their respective colors given three (or more) known data sources with relevant information. A first step towards integrating these various data sources is to model them in a unifying format. In this case, we propose wrapping all of the data sources in the web ontology language format. For structured data sources, our focus for this effort, the OWL wrapping provides an approximate ontology for the given data source.

Once the data sources are wrapped in OWL, the semantics of the user query must be mapped to that of the given data sources. This is the problem of ontology alignment. We will leverage Ontrapro to build approximate alignments between these systems.

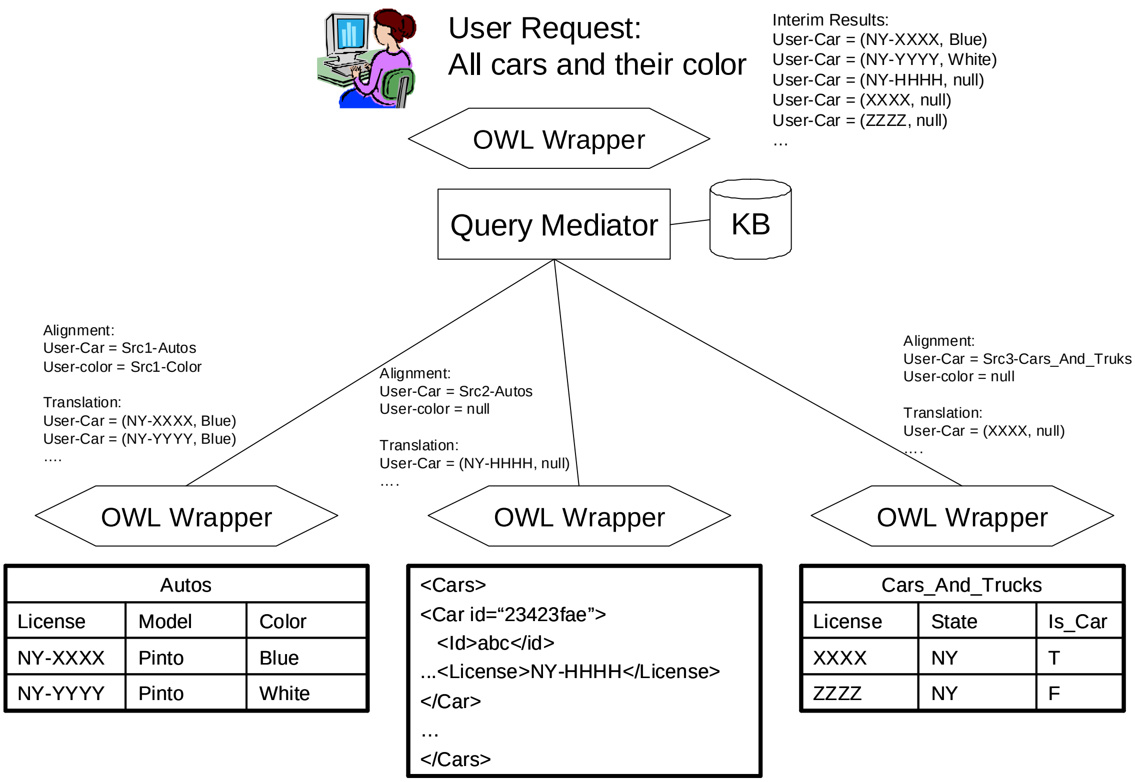

Figure 7 - Federated Search Example

For the data sources in the graphic above, the ontologies align in a fairly straightforward manner. Even in these cases, interoperability is not achieved without significant work in other areas. In the example above, we propose querying each individual data sources based on the user query. The answer will be translated to the user ontology via fuselets transformations and the results will be stored into a knowledge base. It is extremely challenging to determine what information to query from the individual data sources as the combination of information from various sources may lead to an inferred solution. Once the necessary data is stored in the user's local knowledge base, the query can be sent directly to the KB.

In this simple example the initial results produced by the system are incorrect. They include duplicate records and trucks. The idea is to propose solutions to the user and have them help detect problems in the result set. While this example shows problems referring to mapping and entity resolution, we anticipate imprecise results for all areas of semantic interoperability. There are many strategies that could be taken to get user feedback (e.g. showing partial results, asking about an alignment, asking about a co-reference resolution, etc.). The goal is to only burden users based on their integration quality required and their tolerance to interact with the system.

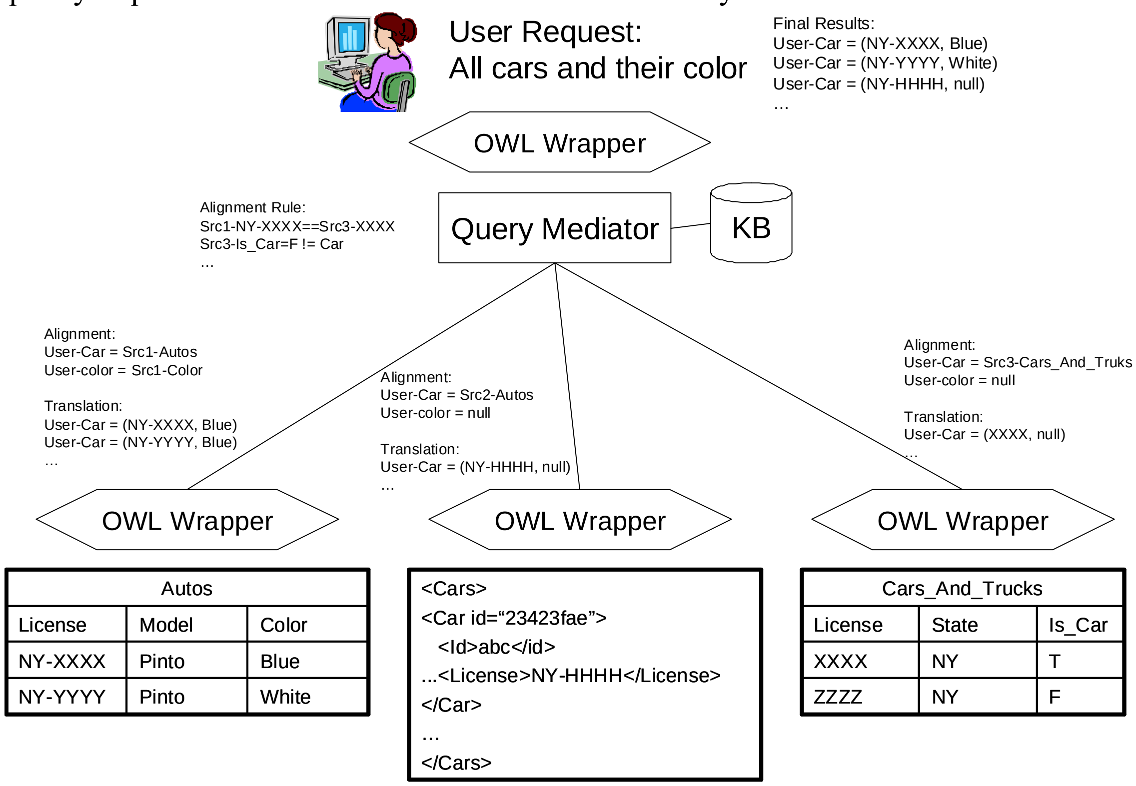

Figure 8 - Federated Search Example Revisited

To implement a similar solution we focused on two notional databases that had related information. You can first align the given data sources. Of particular interest, is determining which ontology or schema should be used for alignment. We choose to allow the existence of a new ontology that could support the merging of the individual ontology structures. One could imagine a case where either of the two original source ontologies were used. Given the existence of a SPARQL query that complied with either an ontology for either source or an overarching ontology, the SPARQL query would query a knowledge base that had a unifying information source.

The multiple source query is a great opportunity to leverage semantic interoperability as it a problem that may support imprecise answers under certain circumstances. Humans are also easily leveraged as they are likely to make the queries and compose the answers. It is quite likely that in this process a human would be willing to answer a couple of requests from the machine or may also recognize results that are either incorrect or incomplete. We believe this area of research would provide a great framework for further semantic interoperability.

Ontology Merging

The third and final demonstration thread developed focused on integrating sources into a unified view of the two sources. The result is essentially a union of the two ontologies and data sources. The hypothesis is that with semantic interoperability technologies, the barrier to solve these problems is significantly lower than without.

Though this approach may be impractical for many problems, many intelligence problems can support the process of integrating all of the available information about a specific topic and putting in a central knowledge base. The advantages of the centralized solution are often necessary and acceptable for certain problems.

To demonstrate the idea, we took information from three ontologies:

- Friends of a Friend (FOAF) – "Who knows who?"
- Group and Membership Ontology – "Who belongs to which group?"
- Financial Ontology – "Which group funds what other groups?"

The goal is the merge these three ontologies and three accompanying data sets. You can then reason about the data to find a suspicious relationship. The end-user will go through a process of aligning two pairs of the three ontologies. They can refine the system generated alignment by rejecting false positives. This rejection step will cause new alignments to potentially be discovered. Finally the user can create a merged ontology which the data can be queried over. Finally, the data must be translated to make the results viewable to the end user.

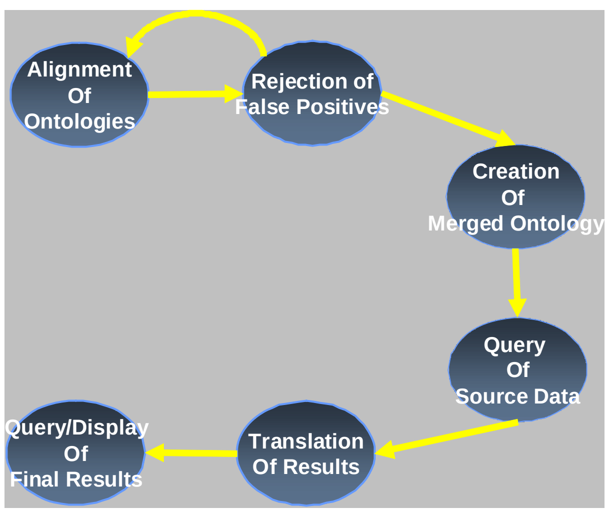

Figure 9 - Ontology Merging Process

The important lesson of this is that there is high value in this space. This is the problem that is most often associated with semantic interoperability. We believe there is value of leveraging the human along with the power of the machine.

Lessons Learned

This and previous semantic interoperability-related efforts have demonstrated a few key lessons that are important to document and takeaway.

Human-in-the-loop is acceptable and in many cases necessary to solve some semantic interoperability problems. The semantic interoperability research community has focused on automated alignment for the past dozen years. This is a good challenge problem because it is very hard and it is quite easy to measure competing approaches against each other. For nearly all practical problems, there are significant benefits on leveraging the expertise of the user. It is the case that there are diminishing returns on their time investment, but there is a quick payoff the user can receive with minimal intervention. The first major benefit is the human can frame the semantic interoperability problem. Rarely is it the case where two entire databases need to be integrated for a given information need. The problem is typically much smaller, and reducing the problem down make it tractable. This framing step can happen explicitly or implicitly through monitoring the context of the problem they are working on. The human can also, of course, provide feedback to the alignment process. If this is done appropriately, it can be done with limited intrusiveness and with maximal value.

Semantic Interoperability solutions that can equivalently be solved by a developer are to be used with caution. In the recent past, many people have attempted to describe semantic interoperability technologies as those that will eliminate the need for developers to solve problems. Though this seems like a noble cause, the tradeoff between adopting an automated solution and using a developer is not sensible for many organizations. Using developers and going through a formal process has its place. Developers are a known commodity that will eventually deliver results. Problems that require semi-automated alignment should focus on situations where the schemas and the data sources used are dynamic.

Not all Semantic Interoperability problems are created equally. Given the current state of the art technologies, there are problems that are better suited for each the automated, semi-automated, and manual semantic interoperability approaches. One dimension of this is correctness. When correctness is high, manual intervention will be required. Some problems will not require precise results and more automated can be used in such cases. One might question this approach, but Internet search is a great example of a problem where the answers to queries are often incorrect but are satisfiable to end consumers.

SI technologies are still not easy enough for most end users; more research needed. The reality with developing semantic interoperability solutions is that is particular challenging to make solutions that are user friendly. This is true for commercial applications such as Microsoft's BizTalk Suite and it is true for all the research software as well. We have stated that the development on CONOPs and user operation in critical for semantic interoperability, but also user tools and paradigms must be improved. Displays for showing massive data sets or schemas are not very user friendly. More work is needed in this area.

Conclusions

We believe there are opportunities for further research in the area of semantic interoperability, but there are certain areas that are richer than others. We believe that further work must be done on applying the semantic interoperability research to real world problems. While the component research continues to improve the application of the research is impractical in many cases. This problem helps further drive disdain in semantic technologies in general.

As part of the effort, we were able to demonstrate the utility of semantic interoperability research and technologies through three demonstration vignettes. We also authored papers describing the work with on being published at an international conference[6].

[6] ATL Ontology Alignment Study Report and D. Chen, J. Lastusky, J. Starz, and S. Hookway. User Guided Iterative Alignment Approach for Ontology Mapping., SWWS 2008.

References

[Bernstein et al, 2006] Bernstein, P.A., Melnik, S., and Churchill, J.E. 2006. Incremental schema matching. In Proc. VLDB, pages 1167–1170.

[Chen et al, 2006] Bi Chen, He Tan, and Patrick Lambrix. “Structure-based filtering for ontology alignment”

[Doan et al, 2002] AnHai Doan, Jayant Madhavan, Robin Dhamankar, Pedro Domingos, and Alon Halevy. “Learning to match ontologies on the Semantic Web”

[Euzenat et al, 2005] Jerome Euzenat, Philippe Guegan, and Petko Valtchev. “OLA in the OAEI 2005 alignment contest”

[Giunchiglia et al, 2004] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. “S-Match: An Algorithm and an Implementation of Semantic Matching”

[Gligorov et al, 2007] Risto Gligorov, Zharko Aleksovski, Warner ten Kate, and Frank van Harmelen. “Using Google Distance to Weight Approximate Ontology Matches”

[Hoshiai et al, 2004] Tadashi Hoshiai, Yasuo Yamane, Daisuke Nakamura, and Hiroshi Tsuda. “A Semantic Category Matching Approach to Ontology Alignment”

[Hughes et al] Todd C. Huges and Benjamin C. Ashpole. “The Semantics of Ontology Alignment”

[Kotis et al, 2004] K. Kotis, G.A. Vouros, and K. Stergiou. “Capturing Semantics Towards Automatic Coordination of Domain Ontologies”

[Le et al, 2004] B.T. Le, R. Dieng-Kuntz, and F. Gandon. “On Ontology Matching Problems – for Building a Corporate Semantic Web in a Multi-Communities Organization”

[Madhavan et al, 2001] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. “Generic Schema Matching with Cupid”

[Meilicke et al] Christian Meilicke, Heiner Stuckenschmidt, and Andrei Tamilin. “Improving Automatically Created Mappings using Logical Reasoning”

[Melnik, 2002] Melnik. S., Garcia-Molina, H., and Rahm, E. 2002. Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching. In Proceedings of the International Conference on Data Engineering, 117-124. San Jose, CA: IEEE Computer Society.

[Melnik et al, 2002] Sergey Melnk, Hector Garcia-Molina, and Erhard Rahm. “Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching”

[Noy et al, 2001] Noy, N.F. and Musen, M.A. 2001. Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In Proceedings of the Workshop on Ontologies and Information Sharing at International Joint Conference on Artificial Intelligence, 63-70. Seattle, WA.

[Qu et al, 2006] Yuzhong Qu, Wei Hu, and Gong Cheng. “Constructing Virtual Documents for Ontology Matching”

[Rahm et al, 2004] Erhard Rahm, Hong-Hai Do, and Sabine Mabmann. “Matching Large XML Schemas”

[Rasgado et al, 2006] Alma Delia Cuevas Rasgado and Adolfo Guzman Arenas. “A language and Algorithm for Automatic Merging of Ontologies”

[Rosse, 2003] Rosse C, Mejino JVL. 2003. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. J Biomed Inform. 36:478-500.

[Sampson, 2005] Ontology Alignment in Agent Systems: Current and future challenges

[Yanosy, 2006] Semantic Interoperability: Net Centric Perspective.